AMENDMENTS TO THE CLAIMS

1.   (Currently Amended) A method for creating a file information database comprising:

scanning a storage server having a directory structure;

collecting data regarding the directory structure;

for each directory of the directory structure, determining whether each member of the directory is a file or subdirectory;

using a first thread to:

traverse the directory structure in a depth first search (DFS) order; and

assign a first unique identification (ID) number to a first determined directory and a second unique ID number to a second determined directory in the directory structure according to a depth first search (DFS) order, wherein the directory ID numbers are assigned while the directory structure is being traversed in the DFS order, and wherein the ID numbers correspond to the DFS order in which the determined directories are traversed;

using a second thread to examine the determined files; and

writing a data structure including the first ID number, the second ID number and a relation between the first directory and the second directory.

2.   (Original) The method of claim 1, wherein scanning and collecting comprise scanning and collecting by using an agent separate from the storage server.

3.   (Original) The method of claim 2, wherein the agent has a first file system, and the storage server has a second file system, and wherein the first file system is different from the second file system.

4.   (Previously Presented) The method of claim 1, wherein the relation indicates that the first directory is an immediate child of the second directory.

5.   (Original) The method of claim 1, wherein assigning further comprises assigning the ID numbers while collecting the data.

6.   (Original) The method of claim 1, wherein writing the data structure further comprises writing the data structure to a database server.

7.   (Previously Presented) The method of claim 4, further comprising:
receiving a request to determine the parent of the first directory; and
referencing the relation between the first directory and the second directory of the
       data structure to determine the parent of the first directory.

8.   (Previously Presented) The method of claim 4, further comprising:
receiving a request to determine an immediate child of the second directory;
searching the data structure to find any relation, including the relation between
       the first directory and the second directory, which indicates that the
       second directory is a parent in said relation; and
determining the immediate child of the second directory based on said any
       relation.

9.   (Previously Presented) The method of claim 4, further comprising:
receiving a request to determine a set of ID numbers of every child of a third
       directory in the directory structure, wherein the third directory is assigned
       a third ID number;
determining a fourth ID number of a sibling of the third directory; and
determining the set of ID numbers between the third ID number and the fourth ID
       number.

10.   (Currently Amended) A machine readable medium having stored thereon executable program code which, when executed, causes a machine to perform a method for creating a file information database, the method comprising:

scanning a storage server having a directory structure;

collecting data regarding the directory structure;

for each directory of the directory structure, determining whether each member of the directory is a file or subdirectory;

using a first thread to:

traverse the directory structure in a depth first search (DFS) order; and

assign a first unique identification (ID) number to a first determined directory and a second unique ID number to a second determined directory in the directory structure according to a depth first search (DFS) order, wherein the directory ID numbers are assigned while the directory structure is being traversed in the DFS order, and wherein the ID numbers depend on the DFS order in which the determined directories are traversed;

using a second thread to examine the determined files; and

writing a data structure including the first ID number, the second ID number and a relation between the first directory and the second directory.

11.   (Original) The machine readable medium of claim 10, wherein scanning and collecting comprise scanning and collecting using an agent separate from the storage server.

12.   (Original) The machine readable medium of claim 11, wherein the agent has a first file system, and the storage server has a second file system, and wherein the first file system is different from the second file system.

-4-

13.   (Previously Presented) The machine readable medium of claim 10, wherein the relation indicates that the first directory is an immediate child of the second directory.

14.   (Original) The machine readable medium of claim 10, wherein assigning further comprises assigning the ID numbers while collecting the data.

15.   (Original) The machine readable medium of claim 10, wherein writing the data structure further comprises writing the data structure to a database server.

16.   (Previously Presented) The machine readable medium of claim 13, further comprising:

receiving a request to determine the parent of the first directory; and

referencing the relation between the first directory and the second directory of the data structure to determine the parent of the first directory.

17.   (Previously Presented) The machine readable medium of claim 13, further comprising:

receiving a request to determine an immediate child of the second directory;

searching the data structure to find any relation, including the relation between the first directory and the second directory, which indicates that the second directory is a parent in said relation; and

determining the immediate child of the second directory based on said any relation.

18.   (Previously Presented) The machine readable medium of claim 13, further comprising:

      receiving a request to determine a set of ID numbers of every child of a third directory in the directory structure, wherein the third directory is assigned a third ID number;

      determining a fourth ID number of a sibling of the third directory; and

      determining the set of ID numbers between the third ID number and the fourth ID number.

19-27.   (Canceled)

28.   (Currently Amended) A method for creating a logical tree comprising:

using a directory walking thread to examine a first directory from a top of a directory queue, and determine a set of children of the first directory;

assigning a depth first search (DFS) identifier (ID) to the first directory, wherein the ~~directory numbers are~~ DFS ID is assigned while the directory structure is being traversed in ~~the~~ a DFS order, and wherein the DFS ID is a unique ID that corresponds to the DFS order in which the first directory is traversed;

examining a set of children of the first directory to determine a first subset of files and a second subset of directories;

placing the first subset of files in a file queue for examination by a file thread; and

placing the second subset on the top of the directory queue.

29-31.   (Canceled)

32.   (Currently Amended) The method of claim 28, ~~wherein~~ further comprising examining the file queue by the file thread, wherein the examining ~~further~~

~~comprises~~includes recording information about a first file ~~taken~~ examined from the file queue.

33.   (Canceled)

34.   (Previously Presented) The method of claim 28, wherein the directory walking thread is hosted by an agent that is separate from the storage server.

35.   (Currently Amended) The method of claim 34, further comprising using ~~an~~ a multi-appliance management application (MMA) to control the agent.

36.   (Previously Presented) The method of claim 34, wherein the directories are hosted by a filer.

37-38. (Canceled)

39.   (Currently Amended) A method for creating a file information database comprising:

scanning a storage server having a directory structure;

for each directory of the directory structure, determining whether each member of the directory is a file or subdirectory;

using a first thread to assign a first unique identification (ID) number to a first determined directory and a second unique ID number to a second determined directory in the directory structure according to a depth first search (DFS) order, wherein the ~~directory~~ ID numbers are ~~chronologically assigned in numerical order~~ assigned while the directory structure is being traversed in the DFS order, and wherein the ID numbers are chronologically assigned in numerical order based on the DFS order in which the directory structure is traversed;

using a second thread to examine the determined files; and

writing a data structure including the first ID number, the second ID number and a relation between the first directory and the second directory.

40.   (Previously Presented) The method of claim 1, wherein a top level directory of the directory structure is assigned an ID of "0" (zero).

41.   (Currently Amended)   A method for storing results of a file walk of a storage server comprising:

performing a file walk of a storage server, wherein performing the file walk includes assigning unique identification numbers to directories of the storage server in a depth first search order during the file walk, and wherein the unique identification numbers correspond to the depth first search order in which the directories are accessed during the file walk;

storing in a data structure indications of the directories of the storage server, the indications stored in association with the unique identification numbers assigned during the file walk; and

traversing the stored indicationsdata structure based on the unique identification numbers to determine relationships between the directories of the storage server.

42.   (New)   The method of claim 41, wherein each child of a given directory has a unique identification number that is greater than a unique identification of the given directory.

43.   (New) The method of claim 1, wherein the data structure further includes at least a portion of the collected data regarding the directory structure.

44.   (New) A method for using multiple threads to perform a file walk of a
storage server comprising:

using a mutual exclusion object to ensure that a directory walking thread and a
file thread do not operate on the same data at the same time;

using the directory walking thread to:

sequentially assign unique identifiers (IDs) to directories in a directory
queue, wherein the unique IDs correspond to an order in which the
directories are assigned the unique IDs; and

when a unique ID is assigned to a directory, signal the mutual exclusion
object that the file thread may continue; and

when the mutual exclusion object indicates that the file thread may continue,
using the file thread to:

examine the members of the directory and place subdirectories of the
directory in the directory queue; and

signal the mutual exclusion object that the directory walking thread may
continue assigning the unique IDs.

45.   (New) The method of claim 44, wherein using the file thread to examine
the members of the directory includes determining whether the members of the
directory are subdirectories or files, and wherein the method further comprises, after the
file thread examines the members of the directory, using the file thread to record data
about the files of the directory.

46.   (New) A system for storing results of a file walk of a storage server
comprising:

an agent to perform a file walk of a storage server, wherein the agent assigns
unique identification numbers to directories of the storage server during
the file walk, and wherein the unique identification numbers correspond to

an order in which the directories are accessed by the agent during the file walk;

a data structure to store indications of the directories of the storage server, wherein the indications are stored in association with the unique identification numbers assigned during the file walk, such that the data structure is traversable based on the unique identification numbers to determine relationships between the directories of the storage server.